

问题 Wk.11.1.1: 观察模型

在这个问题中，我们着眼于定义观察模型。前三个问题要求您根据房间的实际颜色定义机器人在房间中观察到的颜色的条件分布。在解决这个问题之前，请务必阅读设计实验室 11 的讲义。

看看[软件文档分布模块](#)。在整个这个问题中，分布模块已导入，因此您可以使用分布.x。

您可以假设实验室讲义中描述的以下函数和变量已经定义：

- makeObservationModel (走廊颜色, obsDist) --参见实验室讲义。
- 标准走廊--走廊中每个房间的颜色列表，从左到右。在本题中，这是

['白色', '白色', '绿色', '白色', '白色']。

- 可能的颜色--所有可能观察到的颜色的列表。在这个问题中，这是

['黑色', '白色', '红色', '绿色', '蓝色']

在您的答案中，请随意定义您可能需要的任何辅助功能。

第一部分：白色 == 绿色

定义一个观测噪声模型（即给定房间的实际颜色，对观察到的颜色的条件分布），其中白色和绿色是无法区分的，也就是说，当机器人处于白色方块或绿色方块中时，它看到白色的可能性与看到绿色的可能性一样大，但其他所有颜色都能被完美地观察到。此函数应返回一个距离.DDist通过观察到的颜色，给出作为参数传入的实际颜色。

```
def whiteEqGreenObsDist (actualColor):
    pass
```

第 2 部分：白色 <-> 绿色

给出一个观察噪声分布，其中白色总是看起来是绿色，绿色总是看起来是白色，并且所有其他颜色都被完美地观察到。

```
def whiteVsGreenObsDist (actualColor):  
    pass
```

第三部分：嘈杂

定义一个观测噪声分布，其中观察到房间实际颜色的概率为 0.8，看到剩余颜色之一的概率为 0.2（看到其他颜色的概率相同）。所有可能的颜色都位于名为可能的颜色。

```
def noisyObs(actualColor):  
    pass
```

第四部分：观察模型

到目前为止，您定义的观测噪声分布以与位置无关的方式对观测的误差模型进行编码。但最终，我们需要能够获得走廊上特定位置的可能观测（颜色）的概率分布。函数制作观察模型允许我们构建这样一个模型，给定一个颜色列表，例如标准走廊，以及观察噪声分布，如您上面所写的。

输入表达式以创建完整的观测模型标准走廊
（假设它已经定义），观察噪声分布嘈杂的观察（假设它已经被定义）。

```
noisyObsModel = None
```

问题 Wk.11.1.2: 转换模型

在这个问题中，我们来看看如何定义过渡模型。在开始这个问题之前，阅读设计实验室 11 讲义很重要。

看看[软件文档分布模块](#). 吞吐量这个问题，分布模块已导入，因此您可以使用分布.x。

在您的答案中，请随意定义您可能需要的任何辅助功能。

下列有用的函数已经定义：

```
def incrDictEntry (d, k, v) :  
    如果 d.has_key(k):  
        d[k] += v  
    别的:  
        d[k] = v
```

第一部分：生活在甜甜圈上

写一个 `ring` 动力学模型，其中房间 0 连接到房间走廊长度-1。

```
def ringDynamics(loc, act, hallwayLength):  
    pass
```

第二部分：向左滑动

写左滑噪声模型，其中，机器人以 0.1 的概率落在**左边**其标称位置之外，否则落在其标称位置。但是，不应允许它过渡到世界尽头：如果它标称位于最左边的位置，则它应该以 1 的概率停留在那里。

```
def leftSlipTrans(nominalLoc, hallwayLength):  
    pass
```

第 3 部分：嘈杂的过渡

编写一个过渡噪声模型，其中机器人以 0.1 的概率降落在它应该在的位置左侧的一个方格上，以 0.1 的概率降落在它应该在的位置右侧的一个方格上，以 0.8 的概率降落在标称方格上。它不应该被允许从世界的任一端过渡出去；任何与走廊尽头的方格相关的概率都应该与走廊尽头的方格相关。

```
def noisyTrans(nominalLoc, hallwayLength):  
    pass
```

第四部分：转型模型

函数makeTransitionModel在讲义中进行了描述，将动力学和噪声模型结合在一起，为给定长度的走廊构建完整的过渡模型。

输入创建完整过渡模型的表达式标准动力和吵闹的翻译（假设它已经定义）在标准走廊（假设它已经定义了）其长度为 5 个正方形。

```
noisyTransModel = None
```

问题 Wk.11.1.4: 模拟走廊

以下是一些状态估计的手动模拟。

第一部分：完美的传感器，完美的动作

请参阅课程笔记（第 7.7 节）。假设机器人位于一个有 3 个房间的走廊中，中间有一个绿色房间，两边各有一个白色房间。机器人正在尝试估计它在哪个房间。为此，我们将使用状态估计，状态是房间——表示为索引 0 到 2。

我们假设最初我们不知道机器人在哪里，并且所有状态的可能性都是相同的。

我们还假设传感器和运动模型完美，也就是说，运动或传感中没有不确定性。如果命令操作（-1、0 或 1 之一）将使机器人离开走廊边缘，那么机器人将尽可能远地移动，然后停止。

回想一下大写字母 S_t, I_t, O_t 参考时间状态的随机变量 t ，
行动 t ，并观察时间 t 。小写符号， s, i, o 是正常（非随机）变量，表示状态、动作和
 D_s

观察，因此 $\sum_{s \in D_s} Pr(S_0 = s) = 1.0$ 。

我们鼓励你使用分数进行计算；你可以输入分数，例如 5/27，在下面的框中输入。如果输入小数，则需要精确到 0.001 以内。

1. 机器人的先验信念是什么 $B_0(s) = Pr(S_0 = s)$ 对于每个州 s ?

$$B_0(s) = Pr(S_0 = s)$$

2. 首先，机器人进行观察。我们假设它看到“白色”，因为它在白色的房间，没有传感器噪音。所以， $O_0 = \text{white}$ 。我们想知道什么这次观察之后的新的信念状态是什么？

- 首先，弄清楚 $Pr(O_0 = \text{white} | S_0 = s)$ 对于每个州 s

- 然后计算 $Pr(O_0 = \text{white} | S_0 = s) Pr(S_0 = s)$ 对于每个州 s 。请注意，这等同于 $Pr(O_0 = \text{white}, S_0 = s)$

- 现在，计算 $Pr(O_0 = \text{white})$ 。

- 计算观察后的新信念状态；使用条件概率的定义和前两个结果：

$$B'_0(s) = Pr(S_0 = s | O_0 = \text{white})$$

3. 如果我们告诉机器人向右走一个房间，动作状态是考虑状态转换 $I_0 = 1$ ，信仰是什么之后的吗？

$$B_1(s) = Pr(S_1 = s | O_0 = \text{white}, I_0 = 1)$$

4. 现在，假设机器人观察到“绿色”，因为它在绿色房间里，而且没有噪音，
 $O_1 = green$). 这之后的信仰状态会是怎样？

$$B'_1(s) = Pr(S_1 = s | O_0 = white, I_0 = 1, O_1 = green)$$

5. 如果我们考虑到状态转换后告诉机器人立即采取行动 $I_1 = 1$ ，信仰状态会是什么呢？

$$B_2(s) = Pr(S_2 = s | O_0 = white, I_0 = 1, O_1 = green, I_1 = 1)$$

第 2 部分：传感器噪音，动作完美

请参阅课程笔记（第 7.7 节）。

假设机器人位于一个有 3 个房间的走廊中，中间有一个绿色房间，两边各有一个白色房间。机器人正在尝试估计自己在哪个房间。为此，我们将使用状态估计，状态为房间——用索引 0 到 2 表示。

我们假设最初我们不知道机器人在哪里，并且所有状态的可能性都是相同的。

我们假设有 5 种可能的颜色：

（‘黑色’、‘白色’、‘红色’、‘绿色’、‘蓝色’）

我们假设一个噪声传感器，它有 0.8 的概率看到当前房间的正确颜色，有 0.05 的概率看到其他可能的颜色

我们假设完美的运动，动作为 -1、0、1。如果命令的动作会让机器人离开走廊的边缘，那么机器人就会尽可能地移动，然后停下来。

回想一下大写字母 S_t, I_t, O_t 参考时间状态的随机变量 t 吨，
行动 t 吨，并观察时间 t 吨。小写符号， s ，我，哦是正常（非随机）变量，表示状态、动作和

观察，因此 $\sum_{s \in D_s} Pr(S_0 = s) = 1.0$ 。

我们鼓励你使用分数进行计算；你可以输入分数，例如 5/27，在下面的框中输入。如果输入小数，则需要精确到 0.001 以内。

1. 机器人对每个状态的先验信念是什么 s ?

$$B_0(s) = Pr(S_0 = s)$$

2. 机器人看到的分布是什么？也就是说，所有可能的颜色是什么 o ? $Pr(O_0 = o)$ 为了

$$P(O_0 = black) = \text{ } P(O_0 = white) = \text{ } P(O_0 = red) = \text{ }$$

$$P(O_0 = green) = \text{ } P(O_0 = blue) = \text{ }$$

3. 首先，机器人进行观察。假设它看到“白色”。因此，
我们想知道观察之后新的信念状态。

$$O_0 = white)$$

$$B'_0(s) = Pr(S_0 = s | O_0 = white)$$

4. 如果我们告诉机器人向右走 $I_0 = 1$ ，服用后信念状态会怎样
状态转换？回想一下，运动是完美的。

$$B_1(s) = Pr(S_1 = s | O_0 = white, I_0 = 1)$$

5. 现在，机器人看到的分布是什么？也就是说，所有可能的颜色的分布是什么 o ?

$$Pr(O_1 = o)$$

$$P(O_1 = black) =$$

$$P(O_1 = white) =$$

$$P(O_1 = red) =$$

$$P(O_1 = green) =$$

$$P(O_1 = blue) =$$

6. 现在，假设机器人再次看到“白色”，这是之后的 $O_1 = white$ 。信仰状态会怎样
事情吗？

$$B'_1(s) = Pr(S_1 = s | O_0 = white, I_0 = 1, O_1 = white)$$

7. 如果我们告诉机器人向右走，状态转换 $I_1 = 1$ ，服用后信念状态会怎样
该如何考虑？

$$Pr(S_2 = s | O_0 = white, I_0 = 1, O_1 = white, I_1 = 1)$$

8. 如果机器人不是看到“白色”然后向右行驶（如上所述），而是看到“绿色”然后向右行驶，
那么信念状态会是什么？也就是说，

$$Pr(S_2 = s | O_0 = white, I_0 = 1, O_1 = green, I_1 = 1)$$

问题 Wk.11.1.5：模拟走廊： 吵闹的案例

以下是一些状态估计的手动模拟。

不要检查您输入的每一个数字！您将用尽检查次数并导致服务器过载。

请参阅课程笔记（第 7.7 节）。

假设机器人位于一个一维网格中，网格中有 3 个方格，中间有一个绿色方格，两边各有一个白色方格。状态索引以 0 为基数。

我们假设最初我们不知道机器人在哪里，并且所有状态的可能性都是相同的。

我们假设有 5 种可能的颜色：

（‘黑色’、‘白色’、‘红色’、‘绿色’、‘蓝色’）

我们假设一个噪声传感器，它有 0.8 的概率看到当前方块的正确颜色，有 0.05 的概率看到其他可能的颜色

我们还假设噪声运动（动作为 -1、0 或 +1）。采取行动的“名义”结果是向适当的方向移动那么多空间。如果该名义结果位置超出界限，则该名义位置将被“剪裁”为界限内（即，它将是走廊的一端或另一端）。如果有噪声，移动到名义方格的概率为 0.8，位于该方格任一侧的概率为 0.1，即移动不够远或移动太远一个方格。如果噪声结果位置在走廊的一端之外，则与该结果相关的概率将分配给适当的走廊末端。

回想一下大写字母 S_t 、 I_t 、 O_t 参考时间状态的随机变量 t ，
行动 a_t ，并观察时间 t 。小写符号， s ， i ， o 是正常（非随机）变量，表示状态、动作和

观察，因此 $\sum_{s \in D_s} Pr(S_0 = s) = 1.0$ 。

我们鼓励你使用分数进行计算；你可以输入分数，例如 5/27，在下面的框中输入。如果输入小数，则需要精确到 0.001 以内。

1. 机器人的先验信念是什么 $B_0(s) = Pr(S_0 = s)$ 对于每个州 s ?

$$B_0(s) = Pr(S_0 = s)$$

2. 首先，机器人进行观察。假设它看到“白色”。因此，

$$O_0 = white$$

。我们想知道观察后新的信念状态

$$B'_0(s) = Pr(S_0 = s | O_0 = white)$$

3. 如果我们告诉机器人向右走，状态转换 $I_0 = 1$ ，服用后信念状态会怎样
该如何考虑呢？

$$B_1(s) = Pr(S_1 = s | O_0 = white, I_0 = 1)$$

4. 现在，假设机器人在此之后再次看到“白色”？ $O_1 = white$. 信仰状态将会是怎样的

$$B'_1(s) = Pr(S_1 = s | O_0 = white, I_0 = 1, O_1 = white)$$

--	--	--

5. 如果我们告诉机器人向右走，状态转换 $I_1 = 1$ ，服用后信念状态会怎样要考虑进去吗？

$$B_2(s) = Pr(S_2 = s | O_0 = white, I_0 = 1, O_1 = white, I_1 = 1)$$

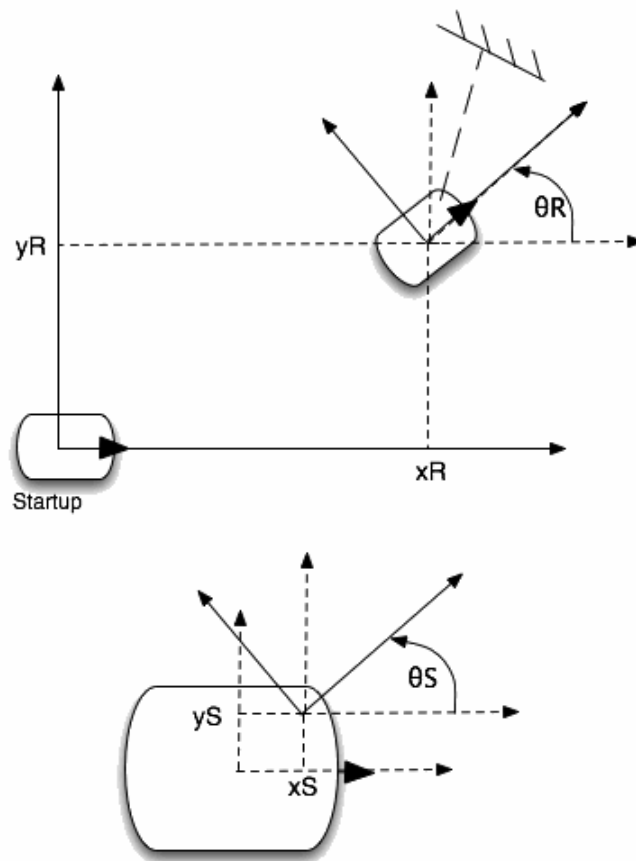
--	--	--

问题 Wk.11.1.6: 声纳命中

在解释机器人上的声纳传感器时，会出现一个计算过程：给定一个声纳的距离测量值，返回一个实例观点（看看[记录实用工具模块](#)），在全局里程计框架中，即测量机器人姿势的同一坐标框架，表示声纳波束从物体上反弹的位置。为此，我们假设声纳波束是从声纳传感器发出的线段。

为了计算这个，我们需要知道位置和方向（[实例姿势](#)），我们需要知道机器人的姿势。

- 机器人上的声纳位置由以下公式给出姿势 (x_S, y_S, θ_S) 在哪里，尺寸和 y_S 是传感器的中心（相对于机器人中心）和西塔是光束与机器人前进方向（机器人鼻子指向的方向）之间的夹角。
- 机器人的姿势是姿势 (x_R, y_R, θ_R) ，如实验室基础设施指南中所述。



在计算这一点时，首先考虑找到命中点相对于机器人的位置，然后计算该点相对于全局里程计框架的位置，这很有用。

这里有一个有用的数学知识。假设你有两个坐标系，分别称为 A 和 B。B 的原点位于 (x_B, y_B) ，相对于 A，B 的 x 轴相对于 A 的 x 轴旋转了 θ_B 。那么，如果我们有一个坐标为 b_x 的点，

相对于坐标系B，我们可以找到该点相对于A的坐标ax和ay，如下所示：

$$\begin{aligned} ax &= xB + \cos(\theta_B) * bx - \sin(\theta_B) * by \\ ay &= yB + \sin(\theta_B) * bx + \cos(\theta_B) * by \end{aligned}$$

请注意，当 θ_B 为零时，这表示 $ax = xB + bx$ 且 $ay = yB + by$ ，这正是我们所期望的。看看[文档姿势变换点在模块中实用工具](#)；它通过将点按 `pose.x` 和 `pose.y` 位移并通过 `pose.theta` 旋转来变换点。

对于调试，您可能会发现绘制测试用例的图很有用，以便了解答案应该是什么。

编写函数 `sonarHit` 给定一个声纳的距离测量值、声纳在机器人上的姿态以及机器人的姿态。它应该返回一个观点（在全局里程计框架中，与测量机器人姿势的坐标框架相同，表示声纳波束从物体上反射的位置。

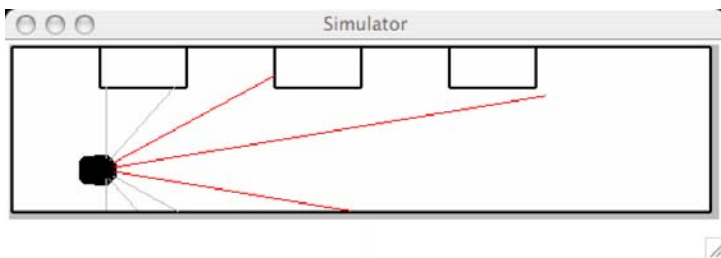
您需要指定实用程序。获取函数和类实用工具。

```
def sonarHit(distance, sonarPose, robotPose):  
    pass
```

问题 Wk.11.1.7: 理想的声纳读数

在已知地图但未知起始位置的情况下，我们把确定机器人在世界上的位置的问题称为**本土化**。我们可以把这个问题定义为状态估计问题。如果我们以定义机器人与世界互动的形式来指定一个模型，[随机SM](#)，并将其输入到状态估计器中，然后状态估计器将不断更新一个概率分布，该概率分布表示根据迄今为止采取的动作和观察，其对机器人位置的信念。

在设计实验室 13 中，我们将构建一个简单的系统，其中机器人开始面向右侧，然后沿着直线向右移动，仅使用其左侧（编号 0）声纳传感器的值来获取信息。



机器人模型的一部分是声纳读数的误差模型。我们的声纳读数误差模型将具有与观察走廊颜色的误差模型类似的结构：如果我们的观察结果完美无缺，我们确定我们期望在每个状态下看到的声纳观察结果，然后将其用作构建可能观察结果分布的基础。

在这个问题中，我们考虑如何计算理想的声纳读数。我们给出：

- 墙段：表示模拟世界中墙壁的线段列表。每个线段都是[工具.LineSeg班级](#)（看看[软件文档](#) 对于此类）。
- 机器人姿势：姿势列表（[实用姿势班级](#)），对应于每个离散机器人状态的中心。
- sonarHit（距离，sonarPose，robotPose）：一个过程，占用来自声纳传感器的线段（分布），传感器在机器人框架中的姿态（表示为实用姿势）以及机器人在全局框架中的姿态（表示为实用姿势），并返回[实用点](#)表示声纳段末端在全局框架中的位置。此函数已定义。
- sonarPose0：声纳传感器 0 相对于机器人的姿态作为以下实例给出实用姿势。
- 声纳：声纳读数的最大有意义值；任何比此值更长的距离都应表示为声纳。
- 观察次数：声纳读数（从 0 到声纳）将被离散化为 0 到观察次数-1。

实施两个程序：

1. 离散声纳离散化声纳读数：它将声纳读数作为输入，并生成整数 bin 索引作为输出。声纳读数的范围在 0 到声纳。这个范围应该离散化为一个固定的数字，观察次数，箱体。请注意，箱体索引从 0 到观察次数-1。任何声纳读数

大于声纳应放入最后一个垃圾箱。

我们强烈建议您画一幅图，将 0 到 1 之间的间隔分为三个区间。考虑一下当输入 0.32 和 0.34 时您希望返回什么答案。以下是一些关于 Python 的有用知识：

- 这圆形的过程接受一个浮点数并返回最接近的整数，作为浮点数。因此，圆形 (2.8) 返回 3.0 和圆形 (2.1) 返回 2.0。
- 这整数过程接受一个浮点数并返回其整数部分。因此，int (2.1) 返回 2。

2.理想阅读获取描述世界的墙壁片段列表和与系统的每个状态相对应的机器人姿势列表，并返回声纳 0 的离散化理想声纳读数列表，每个状态一个. 对于每个状态：

- 确定从声纳传感器 0 当前所在位置开始，沿着传感器方向延伸的线段
长度声纳。
- 找到该线段与世界上任何距离声纳传感器 0 最近的墙段的交点。
- 计算理想的声纳读数，即从传感器到最近交点的距离。您可以使用路口方法

[工具.LineSeg](#)，返回实用点两条线段之间的交点（如果有）；它返回错误的如果没有。如果没有交点，那么理想的声纳读数是声纳. 计算理想声纳读数的离散值。返回离散理想读数列表，每个状态一个。

○

在 Idle 中调试此问题并将您的答案粘贴到这里。

一个测试用例是：

```
def wall((x1, y1), (x2, y2)):
    返回 util.LineSeg(util.Point(x1,y1), util.Point(x2,y2)) wallSegs = [wall((0, 2), (8,
    2)),
                                墙((1, 1.25),(1.5, 1.25)), 墙((2, 1.75),(2.8,
                                1.75))]
机器人姿势 = [util.Pose(0.5, 0.5, 0), util.Pose(1.25, 0.5,0),
               util.Pose(1.75, 1.0, 0), util.Pose(2.5, 1.0, 0)]
```

```
sonarMax = 1.5
numObservations = 10
sonarPose0 = util.Pose(0.08, 0.134, 1.570796)

def discreteSonar(sonarReading):
    pass

def idealReadings(wallSegs, robotPoses):
    pass
```